

Baud rate variations effect on virtual channel based on PIC microcontroller

Sarmad K. Ibrahim^{*1}, Hussein M. Hathal², Riyadh A. Abdulhussein³

Mustansiriyah University, College of Engineering, Baghdad, Iraq

*Corresponding author, e-mail: eng_sarmadnet@uomustansiriyah.edu.iq¹,
hussein.m.hathal@uomustansiriyah.edu.iq², riyadh@uomustansiriyah.edu.iq³

Abstract

Recent year in the world the real applications, usually needed only a few key features of Universal Asynchronous Receiver Transmitter (UART). It is a type of serial communication protocol, which improves the problem of parallel communication and develops effectively in several services. This paper presents a transceiver system based on PIC microcontroller. It also presents software designs to transmit and receive data through the virtual channel. The system is designed to study the effects of baud rate variations between transmitter and receiver for noise and noiseless AWGN channel. The system has been simulated by Proteus simulator version 8.1, and then tested successfully at baud rates (2400, 4800, 9600, and 19200) bps. Simulation results show that the error rate has zero values at the desired baud rate value, and also, at the adjacent values. Thus, the zero level of error rate is increased by increasing baud rate values, which fixed by the transmitter and vice versa.

Keywords: baud rate (BR), microcontroller-based system, proteus simulator version 8.1., RS232 serial communication, UART

Copyright © 2019 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Wireless communication' was first proposed during the 19th century and has since undergone a huge development. Wireless communication is a vital medium for transmitting information without using cables, wires or similar electronic conductors, but making use of electromagnetic waves like RF, IR, satellite communication, etc. Currently, wireless communication includes many devices or technologies, including computers, smartphones, tablets, or laptops. The asynchronous serial communication shows many advantages like high reliability, few transmission lines, long transmission distances, and hence, is used for exchanging data between the computers and other peripherals. This asynchronous serial data communication is generally implemented with the help of a Universal Asynchronous Receiver Transmitter (UART) [1-4]. The UART helps in carrying out a full-duplex communication with the serial link and is widely used for data communication and in control systems. However, the UART applications only use some of its important features [5-8].

In today's world, data communication is an integral component of electronic systems. The commonly used data communication technique is the asynchronous serial data transmission, which is commonly employed in microcontroller-based applications requiring multiple communication rates[9]. Baud rate refers to the rate at which the bits are transmitted. In other words, the baud rate describes the rate at which the data is transmitted within the communication channel[10-12]. Proper communication needs a similar value for the reception and transmission baud rates [1]. Some of the common baud rates include 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600 and 115200 bits/s [1].

The UART controller is an important part used in the serial communicating subsystem of the computer. UART accepts the data bytes and sequentially transmits all the individual bits. At the receiver, another UART assembles all the receiving data bits into the complete data bytes. Such a serial transmission system is used in modems and for non-network communication amongst the terminals, computers and other such devices.

The asynchronous serial communication is generally implemented with the help of a UART [1]. UART helps in carrying out a full-duplex communication in the serial link and is widely used for data communication and control systems. UART consists of the receiver (RX) and

transmitter (TX). The data transmission is controlled by obtaining the data in the parallel structure and thereafter, with the help of the UART, serially transmits the data. On the other hand, the receiver detects this transmitted data, receives the serial transmission, and stores the received data to the parallel format. Universal Asynchronous Receiver Transmitter, present in the receiver, converts the serial data in the parallel format. Such serial communication prevents signal distortion, and data transfer can successfully take place between the systems having a long distance between each other. The Universal Asynchronous Receiver Transmitter serial module can be divided into three different subpart-modules: receiver, transmitter, and a generator of baud rate [2]. Baud rate generator is first part that refers to a type of frequency divider and produces local clock signals. The baud rate frequency factor is determined with the help of the clock frequency of the system and the necessary baud-rate. The factor of frequency is used for dividing factor. In the UART-based data transmission systems, after establishing the baud rate, the internal clocks of the receiver and the transmitter are set at similar frequencies. TXD is the UART output or the transmitter site, while the RDX represents the receiver or the UART input site. The UART's receiver model receives the signal in series at the RDX site and converts the signal into a parallel format. Thereafter, the transmit module of the UART converts this data into series bits based on its frame format and then, transmits the data bits through the TXD [2]. Microcomputers are second component includes a microprocessor, Input-Output (I/O) device, and programmer and data memory. Some of the microcomputer systems also include other components like counters, timers, and the analog-to-digital converters. Hence, the microcomputer system ranges from a big computing device with printers, floppy disks, and hard disks to a small single-chip embedded system [13]. One of the most popular microcontrollers includes PIC16-series microcontrollers. Though they are excellent for general use, they also have many limitations. Some of these limitations include 1) Limited programming and data memory, 2) Primitive interrupt structure, and 3) Same interrupt vector is shared between all the interrupt sources. The PIC16-series microcontrollers are unable to provide direct support to the advanced peripheral interfaces like CAN bus, Universal Serial Bus (USB) devices, etc. Furthermore, interfacing with these devices proves to be difficult. These microcontrollers have a limited instructional manual and do not contain instructions for even the basic mathematical features like multiplication and division. Branching instructions are simple and combine the skip and the go-to instructions [13]. PIC18-series microcontrollers are designed by Microchip Inc., and they are mainly used in the high-density, high-pin-count, and complex applications. These microcontrollers offer cost-effective solutions for the general applications that are written in the C-programming language, use the Real-Time Operating System (RTOS) and need a complicated communication protocol stack like CAN, USB, TCP/IP, or ZigBee. The PIC18F microcontroller devices provide a flash program memory ranging between 8-128 kB, data memory ranging between 4-256 kB, operate in the range of 2-5 volts and have a speed ranging from DC to 40 MHz [13].

AWGN is last component that refers to random signals or processes having a flat power spectral density. They contain signals with equal power in the fixed bandwidth at any of the center frequencies. This noise is helpful for measuring and testing the wideband communication circuits. The Additive White Gaussian Noise (AWGN) is commonly used in all the communication channels and is described as the statistically random radio noise having a characteristic wide frequency range amongst all the signals in a communication channel [14].

2. Related Work

Here, the authors have shown that UART was the programming microchip device that controlled the interface between the computer and its peripheral devices. It is a very popular serial data communication circuit. The serial transmission process depends on the shift registers principle. There are 2 forms of different serial transmissions, i.e., Synchronous and the Asynchronous systems. In the synchronous serial communication, the TX and the RX have to transmit at the same time. Furthermore, the asynchronous transmission helps in the transmission of data without sending any clock signals to the RXs. The UART modules are satisfied in the asynchronous data transmission with the help of the VHDL design language. Very stable and reliable results can be obtained. Furthermore, the design displays high flexibility with the reference values and better integration [15]. Many systems have been presented in this study, including high data collection and speed control systems, which depend on the Peripheral Component Interconnect (PCI) and the multi-digital signal processing systems. First-in first-out

(FIFO) was used for communicating between the low and high-speed component, or for communicating between the sub-controllers. First-in first-out (FIFO) is an essential component of such systems and acts as a bridge between several devices. In this study, the authors developed a controller which contained the FPGA-based asynchronous FIFO as its most significant component. This FIFO helped in determining the various features of the controller and could be used for completing the communication in the serial or the parallel ports [16].

Here, the authors have also used an 8 common data bit number, which corresponded to one byte of data. A general ASCII code in the communication device uses only seven LSBs, while the MSB was zero. When the UART was configured in eight data bits, one stop bit, and zero parity bit, the received data was in the form of 0-dddd-ddd 0-1, where d was the data bit and could be either zero or one. Assuming that UART had a configuration of eight data bits, one stop bit, and zero parity bit, the baud rate was generally 4800, 9600, or 19,200 bauds [17]. This circuit produced a Frequency Division as it is able to divide the input frequency by the factor of 2. The Frequency Divider divides the frequency based on the requirements of the system. Hence, the authors used the UART with a frequency divider as this ensured that no other device had to be attached to the system for dividing the frequency. In one study, [18] the authors implemented the two-directional shift converter process with the First in first out and the UART circuit blocks in an FPGA device with the help of the Verilog HDL language that was used in the embedded system converter RS232 connected to the USB. Also, an eight-bit UART module design which was based on the Verilog HDL was employed in [19]. Such a design automatically identified the addresses of every character. The authors also used a VLSI design and passed the data between their proposed 9-bit UART modules with the host CPU. This design consisted of the transmitter, receiver, Prescaler modules, and the asynchronous FIFOs. In another study [20], the researchers developed the process for software implementation of the UART with the help of the shift register. They aimed to obtain a UART-core that could be used as the module for implementing a big system, irrespective of the type of implementation platform selected. Another study used USBs in PCs without serial interfaces [21]. The authors implemented a universal protocol in their Hardware Description Language (HDL) for the ICs, while they used the MATLAB program for the PC. Their design was easily adaptable to the structure and requirements of any specific IC. Authors [22] used the design module for developing a 60-Gb/s nonreturnable-to-zero transceiver having an adaptive equalization and a baud-rate Clock and Data Recovery (CDR). Their design employed a full equalization front end having a per-path adaptation and a per-sampler offset calibration which enabled the operation at a speed of 60-Gb/s over the realistic channels. In [23], the researchers proposed and developed a cost-effective transmitter-based self-calibration system for the frequency response and IQ time delay, which used only the low-bandwidth components. The frequency response correction and the sub-ps timing correction enabled the transmission of 400GE, 66Gbaud DP-16QAM, and a 44Gbaud DP-64QAM. Also, a review of the recent studies in the field of high baud rate signal generation or detection was published in [24]. The authors investigated the various advanced algorithms that were applied at the transmitter and receiver sides for the signal pre-or post- equalization and compensation, respectively. In [25], the authors presented a serial communication design describing the 8051 microcontroller-based system and the PC interconnection with the help of the RS232 design rules. This study also presented a software design for effective data transmission between the systems.

3. Hardware System Design

In this section, a hardware simulation will be implemented and discussed. Transmitter and receiver circuits are implemented and simulated by using Proteus simulator version 8.1 since these circuits are built by PIC 16F877A microcontroller chip. One 16F877A chip is used for transmitter circuit, and another one is used for the receiver circuit. A noise signal is generated by a separate PIC chip. The description of the whole system is shown in Figure 1.

3.1. Software System Design

The software code of both transmitter and receiver systems is written based on Mikro C software package version 6.1 and then simulated by Proteus simulator version 8.1. The following steps of the algorithm can describe the transmitter system algorithm:

- Initialize I/O ports.

- Do recursive loop for sending signal regularly.
- Begin to send data signal after complete authentication with the receiver.
- Repeat step 3 until sending all data signal.
- End program.

The transmitter algorithm is implemented by the following flowchart, as shown in Figure 2.

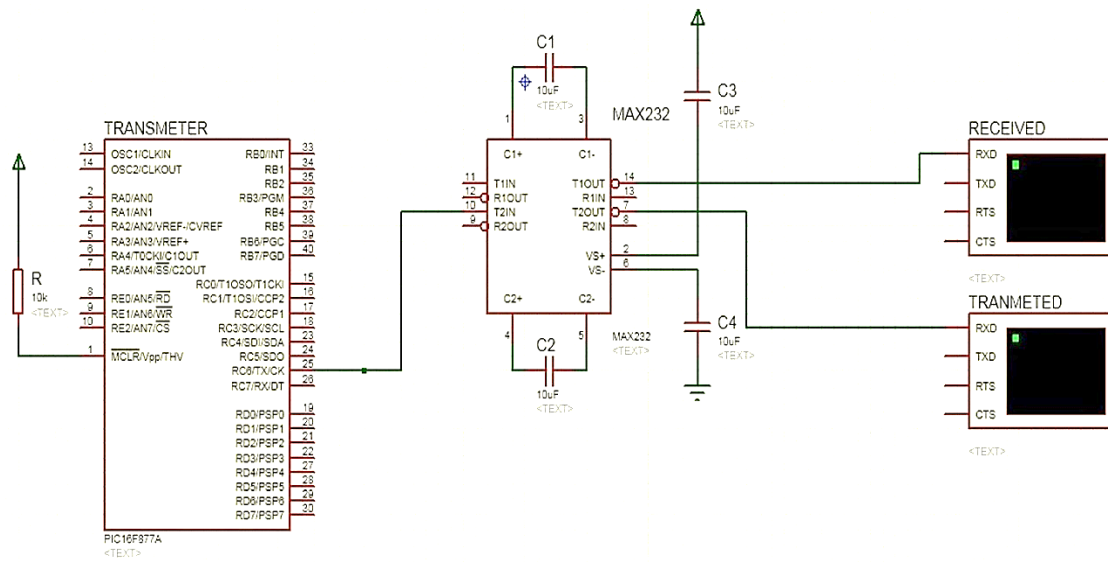


Figure 1. Hardware system design

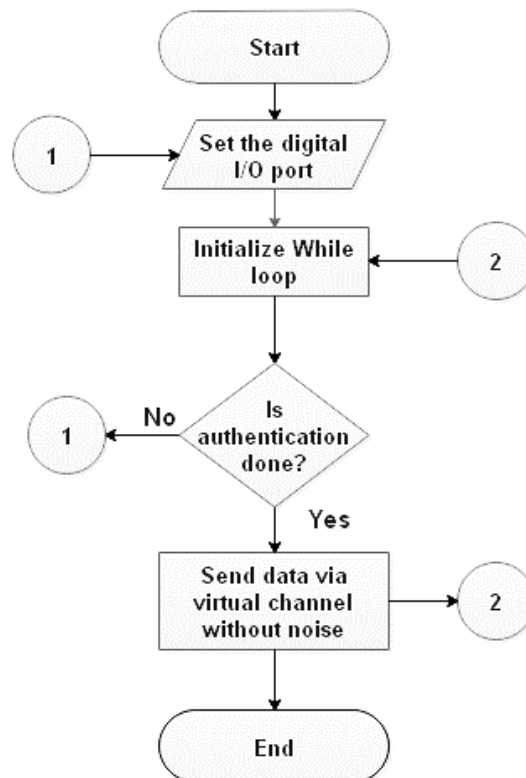


Figure 2. Sending data via virtual channel without noise

The following steps describe the additive white Gaussian noise (AWGN) algorithm:

- Initialize I/O ports
- Do recursive loop to send AWGN noise in a random manner.
- Repeat step 2 infinitely.

The AWGN noise algorithm is implemented by the following flowchart, as shown in Figure 3.

The following steps describe the receiver system algorithm:

- Initialize I/O ports
- Prepare the recursive loop to ensure that authentication from the receiver is done regularly.
- If there is no receiving data, end the program, else, print the receiving data.
- End the program.

The receiver algorithm is implemented by the following flowchart, as shown in Figure 4.

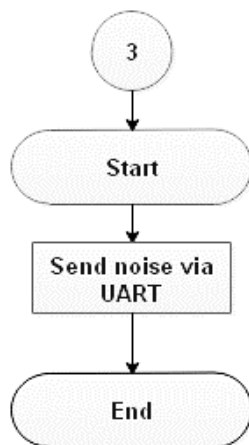


Figure 3. Sending noise via virtual channel

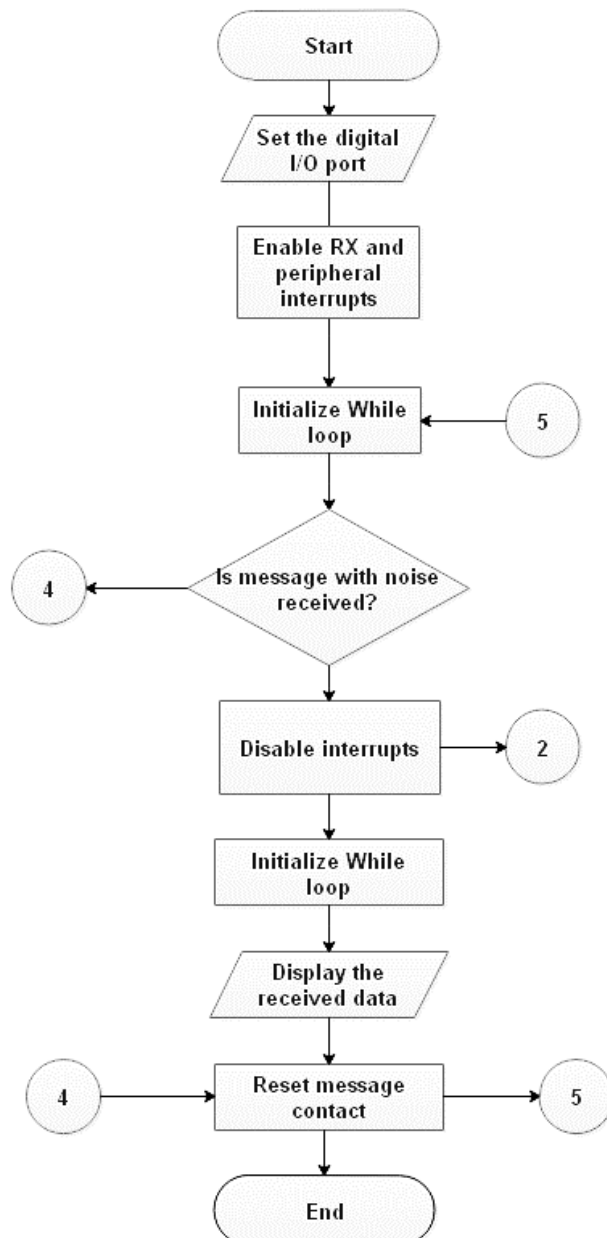


Figure 4. Receiving data via virtual channel without and with noise

4. Simulation Results and Discussion

In order to study the effects of baud rate variations on receiving data signal, we fixed four baud rate values, i.e. (2400, 4800, 9600, and 19200) bps at the transmitter system, while the baud rate values of the receiver are varied in range (1000 to 35000) bps. At receiver system, the error rate is measured for all baud rate values. Also, the study is done for the system with and without existing AWGN noise signal. Case 2400 bps: In this case, the transmitter system is fixed at 2400 bps. Figure 5 shows that the error rate has zero values around baud rate value (2400 bps), and has non-zero values at any value of baud rate except (2400 bps). In the case of receiving signal with existing of AWGN noise, the error rate has a minimum value at the value of (2400 bps) among all values of baud rate, as shown in Figure 5. Case 4800 bps: In this case, the transmitter system is fixed at 4800 bps. Figure 6 shows that the error rate has zero values around baud rate value (4800 bps), and has non-zero values at any value of baud rate except (4800 bps). In the case of receiving signal with existing of AWGN noise, the error rate has a minimum value at the value of (4800 bps) among all values of baud rate, as shown in Figure 6.

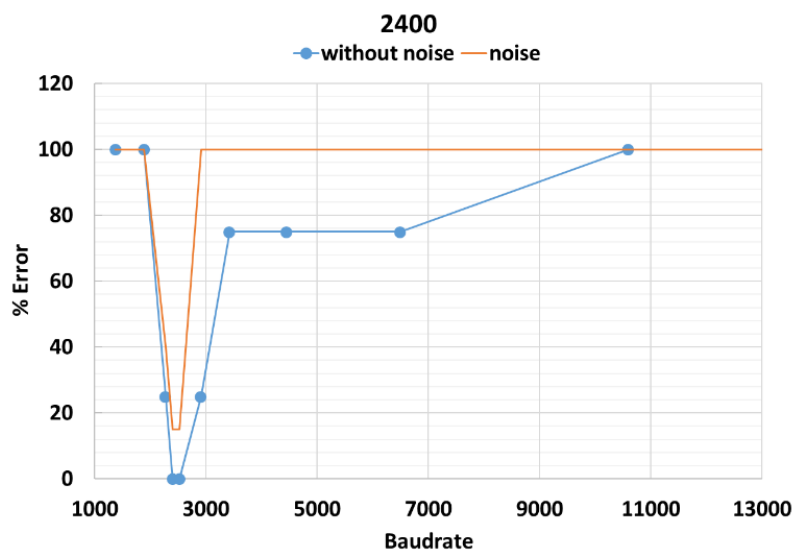


Figure 5. Error rate at (2400 bps)

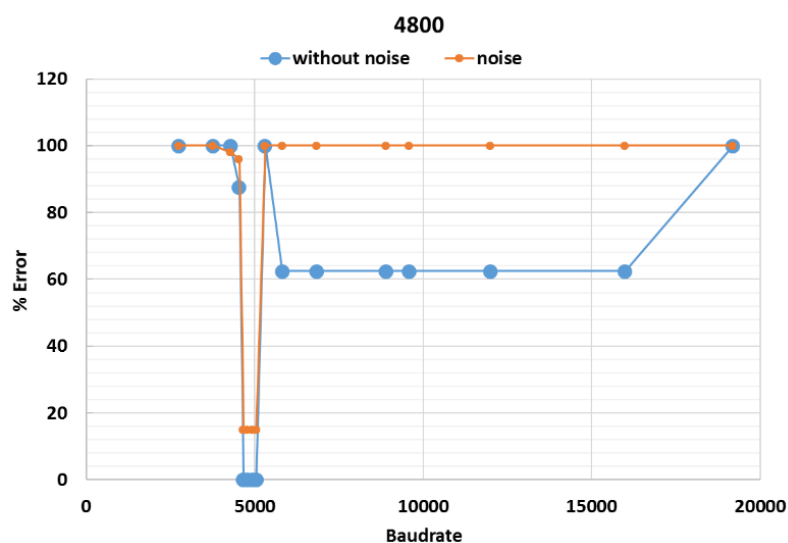


Figure 6. Error rate at (4800 bps)

Case 9600 bps: In this case, the transmitter system is fixed at 9600 bps. Figure 7 shows that the error rate has zero values around baud rate value (9600 bps), and has non-zero values at any value of baud rate except (9600 bps). In the case of receiving signal with existing of AWGN noise, the error rate has a minimum value at the value of (9600 bps) among all values of baud rate, as shown in Figure 7.

Case 19200 bps: In this case, the transmitter system is fixed at 19200 bps. Figure 8 shows that the error rate has zero values around baud rate value (19200 bps), and has non-zero values at any value of baud rate except (19200 bps). In the case of receiving signal with existing of AWGN noise, the error rate has a minimum value at the value of (19200 bps) among all values of baud rate, as shown in Figure (8). As shown in the four cases, the average error rate is increased as increasing baud rate values that fixed at the transmitter. Since the average error rate for case (19200 bps) is larger than the average error rate of the case (9600 bps). Also, the average error rate for case (9600 bps) is larger than the average error rate of the case (4800 bps) and so on. As shown in figures, the error rate has a wider range of zero values for larger baud rate values which fixed at the transmitter. Thus, the error rate for case (19200 bps) has a wider range of zero values than error rate case (9600 bps). Also, the error rate for case (9600 bps) has a wider range of zero values than error rate case (4800 bps) and so on. Therefore, the case (2400 bps) has a narrower range of zeros value than other cases. Thus, simulation results show slight error in receiving signal for baud rate variations as compared with [25].

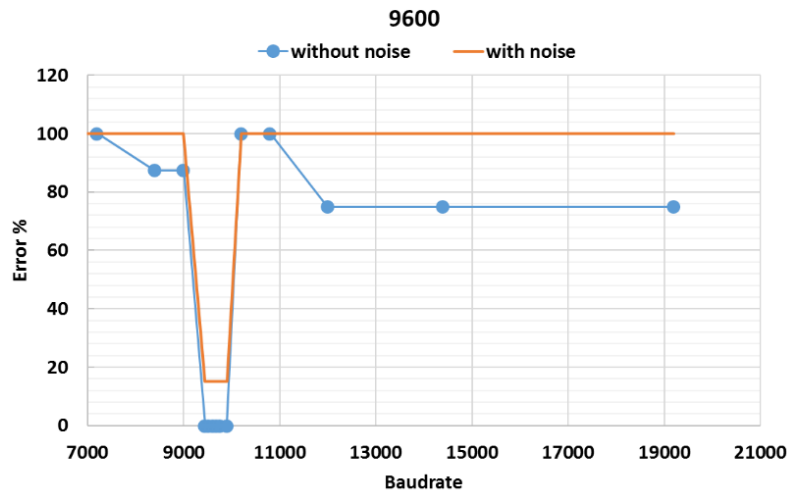


Figure 7. Error rate at (9600 bps)

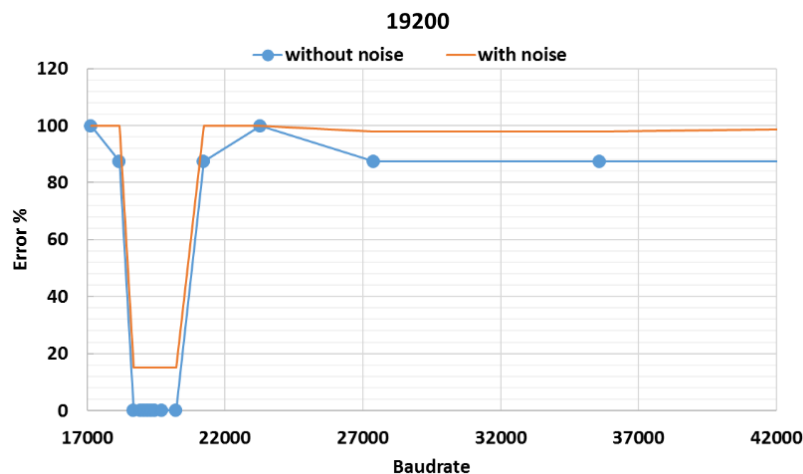


Figure 8. Error rate at (19200 bps)

5. Conclusions

A transmitter system based–PIC microcontroller is implemented and simulated in Proteus simulator version 8.1. The system is designed and tested to send and receive data signals at different values of baud rates. The average error rate is increased for larger values of baud rate that is fixed at transmitter system and vice versa. At larger values of baud rate, the error rate is still continuous to zero levels for not only at the desired baud rate value, but it exceeds to the adjacent values of baud rate, which improve the reliability of received the data signal.

Acknowledgements

The authors would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) Baghdad-Iraq for its support in the present work.

References

- [1] P Mukherjee, PA Nageswara Rao. Automatic Baud Rate Detection using Programmable System on Chip (PSoC). *IJECET*. 2012; 3(2): 130-140.
- [2] P Kedia, NN Mandaogade. A Review Paper on Implementation of UART Controller with Automatic Baud Rate Generator using FPG. *International Journal of Advance Research in Computer Science and Management Studies*. 2014; 2(1): 280-283.
- [3] Liakot Ali, Roslina Sidek, Ishak Aris, Alauddin Mohd. Ali, Bambang Sunaryo Suparjo. Design of a micro - UART for SoC application. *Computers and Electrical Engineering*. 2004; 30(2): 257– 268.
- [4] HU Hua, BAI Feng-e. Design and Simulation of UART Serial Communication Module Based on Verilog–HDL[J]. *Computer and Modernization*. 2008: 8.
- [5] A Kaur, A Kaur. An Approach For Designing A Universal Receiver Transmitter (UART). *International Journal of Engineering Research and Applications (IJERA)*. 2012; 2(3): 2305-2311.
- [6] R K.Agrawal, Vivek Ranjan Mishra. *The Design of High-Speed UART*. Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013). India. 2013: 388-390.
- [7] N Patel Vatsalkumar Patel, Vikaskumar Patel. *VHDL Implementation of UART with Status Register*. In International Conference on Communication Systems and Network Technologies. India. 2012: 750-754.
- [8] GBW Iti Aggarwal, S Gaba. *Synthesis and Implementation of UART using VHDL Codes*. In International Symposium on Computer, Consumer and Control. Taiwan. 2012: 1-3.
- [9] MNAM Alias, SN Mohyar, MN Isa, A Harun, AB Jambek, SAZ Murad. Design and analysis of dedicated Real-time clock for customized microcontroller unit. *Indonesian Journal of Electrical Engineering and Computer Science*. 2019; 14(2): 796-801.
- [10] P Gupta, P Chaudhary. A Review Paper on Design And Simulation of Universal Asynchronous Receiver Transmitter on Field Programmable Gate Array Using VHDL. *International Journal of Advance Research in Science and Engineering (IJARSE)*. 2014; 3(11): 62-66.
- [11] P Shrivastava, S Sharma. Design and simulation of 16 Bit UART Serial Communication Module Based on VHDL. *International Journal of Emerging Technology and Advanced Engineering*. 2014; 4(2): 72-78.
- [12] Amanpreet Kaur, Amandeep Kaur. An approach for designing a universal asynchronous receiver transmitter (UART). *International Journal of Engineering Research and Applications (IJERA)*. 2012; 2(3): 2305-2311.
- [13] D Ibrahim. *Advanced PIC Microcontroller Projects in C*. from USB to RTOS with the PIC 18F Series. Newnes. 2011.
- [14] HM Hathal, RA Abdulhussein, SK Ibrahim. Lyapunov Exponent Testing for AWGN Generator System. *Communications and Network*. 2014; 6(4): 201-208.
- [15] F Yi-yuan, C Xue-jun. *Design and Simulation of UART Serial Communication Module based on VHDL*. 3rd International Workshop on Intelligent Systems and Applications. China. 2011: 1-4.
- [16] S Yu, L Yi, W Chen, and Z Wen. *Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA*. Second IEEE Conference on Industrial Electronics and Applications. China. 2007: 2633-2638.
- [17] B Mahure and R Tanwar. UART with Automatic Baud Rate Generator and Frequency Divider. *Journal of Information Systems and Communication*. 2012; 3(1): 265-268.
- [18] NF Jusoh, A. Ibrahim, MA. Haron, F Sulaiman. *An FPGA Implementation of Shift Converter Block Technique on FIFO for UART*. IEEE International RF and Microwave Conference (RFM 2011), 12th-14th December 2011.Malaysia. 2011: 320-324.

- [19] NF Mahat. *Design of a 9-bit UART Module Based on Verilog*. IEEE-ICSE2012 Proc. Malaysia. 2012: 570-573.
- [20] B Roy. *Platform-Independent Customizable UART Soft-Core*. Third International Conference on Intelligent Systems Modelling and Simulation. IEEE. Malaysia. 2012: 692-694.
- [21] JH Mueller, MHN Moghaddam, B Mohr, S Strache. *An Adaptable UART Based Configuration and Read-out Interface for IC Prototypes*. PRIME 2012. Germany. 2012: 1-4.
- [22] J Han, N Sutardja, Y Lu. Design Techniques for a 60-Gb/s 288-mW NRZ Transceiver with Adaptive Equalization and Baud-Rate Clock and Data Recovery in 65-nm CMOS Technology. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*. 2017; 52(12): 3474 – 3485.
- [23] CRS Fludger, TDuthel, PHermann, T Kupfer. *Low-Cost Transmitter Self-Calibration of Time Delay and Frequency Response for High Baud-Rate QAM Transceivers*. Optical Fiber Communication Conference. America. 2017: 1-3.
- [24] J Zhang, J Yu, H Chien. Advanced Algorithm for High-Baud Rate Signal Generation and Detection. Optical Fiber Communication Conference. America. 2017: 1-3.
- [25] Al Majeed, HJ Neamaa, RA.AH AL-Helali. *Effectiveness of Baud Rate Variation on the Microcontroller Based System and PC Serial Communication*. 2nd Scientific Conference on Information Technology: Applications and Horizons. Iraq. 2010: 1-14.